

Контрольные вопросы:

Лабораторная работа №4. Рефакторинг.

Цель. Научиться применять приемы рефакторинга, касающиеся реорганизации функций и данных, реорганизации условных выражений, задач абстракции.

Контрольные вопросы:

1. Приведите опр. рефакторинга. Что отличает рефакторинг от оптимизации производительности?
Рефакторинг - изменение во внутренней структуре программного обеспечения, имеющее целью облегчить понимание его работы и упростить модификацию, не затрагивая наблюдаемого поведения.

Отличие рефакторинга от оптимизации:

- При рефактинге код становится понятнее; при оптимизации производительности -, в основном, гораздо сложнее для восприятия.
- При рефактинге, в основном, код становится менее эффективен по памяти и по времени.

при оптимизации. - наоборот.

2. Укажите цели рефакторинга.

- Рефакторинг упрощает дальнейшее проектирование;
- Рефакторинг облегчает понимание структуры;
- Рефакторинг повышает читаемость кода;
- Рефакторинг позволяет быстрее писать программы.

3. Когда следует проводить рефакторинг, а когда он не нужен?

Применять:

- правило "Тех страйков" - если пишешь, в 3 местах дублируешь код, применяешь рефакторинг;
- при добавлении новой функции;
- когда необходимо исправить ошибку;
- во время Code Review (анализ кода в команде).

Не применять:

- когда код слишком запутан (уже написан);
- когда код неработоспособен;
- когда близится дата сдачи проекта.

4. Приведите список признаков проблемного кода ("злых" кода).

Duplicated Code, Long Method, Large Class,
Long Parameter List, Divergent Change,
Shotgun Surgery, Feature Envy, Data Clumps,
Primitive Obsession, Switch Statements, Parallel
Inheritance Hierarchies, Lazy Class, Speculative
Generality, Temporary Field, Message Chains,
Middle Man, Inappropriate Intimacy, Alternative
Classes with Different Interfaces, Incomplete
Library Class, Data Class, Refused Request,
Comments

5. Перечислите принцип рефакторинга, касающийся
составления методов. Опишите принцип "выделение
метода" и "вытравливание метода"

Составление методов:

- выделение метода;
- вытравливание метода;
- вытравливание временной переменной;
- замена временной переменной базовым методом;
- введение поясняющей переменной;
- расчленение временной переменной;

- удаление присваиваний параметрам;
- замена метода объектам методов;
- замещение алгоритма.

Введение метода:

Описание: Это участок кода, который можно группировать;

Действие: Поместить участок кода в метод, название которого отвечает назначению.

Встраивание метода:

Описание: Тело метода столь же точно, как и его название.

Действие: Поместить тело метода в код, код его вызывает, и удалить метод.

6 Привет редактору, касающиеся перемещения функций между объектами. Опишем три типа "перемещения метода" и "перемещение поля".

Перемещение функций между объектами:

- перемещение метода; - встраивание класса;
- перемещение поля; - сокращение дублирования;
- выделение класса; - удаление посредника;

- введение внешнего метода;
- введение локального расширения.

Перемещение метода.

Проблема: метод используется в другом классе, больше, чем в собственном.

Решение: создать новый метод в классе, который использует его больше других, и перенести туда код из старого метода. Код оригинального метода превратить в обращение к новому методу в другом классе либо убрать его вообще.

Перемещение поля:

Проблема: поле используется в другом классе больше, чем в собственном.

Решение: создать поле в новом классе и перенаправить к нему всех пользователей старого поля.

7. Терминиста приема редакторства, касающиеся реорганизации данных.

Реорганизация данных:

- самонаконтрастирующее поле;
- замена значений объектов;

- замена значения констант (и наоборот);
- замена массива объектов;
- дублирование выделенных данных;
- замена односторонней связи на двустороннюю (и наоборот);
- замена математического знака символьной константой;
- инкапсуляция полей;
- инкапсуляция коллекций.

8. Переименование принципа рефакторинга, касающегося упрощения входов методов

Упрощение входов методов:

- переименование метода;
- добавление / удаление параметра;
- разделение запроса и модификатора;
- параметризация метода;
- замена параметра входом метода;
- введение объекта параметров;
- удаление сеттера;
- сокращение метода.

9. Перечислите приемы редактора, касающиеся реорганизации условных выражений.

Реорганизация условных выражений:

- декомпозиция условного оператора;
- консолидация условного выражения;
- консолидация дублирующихся условных фрагментов;
- удаление управляющего потока;
- замена вложенных условных операторов значимым оператором;
- замена условного оператора полиморфизмом;
- введение объекта NULL.

10. Перечислите приемы редактора, касающиеся рефакторинга задач обобщения.

Рефакторинг задач обобщения:

- подъем кода / Подъем метода;
- спуск кода / Спуск метода;
- выделение подкласса;
- выделение родительского класса;
- выделение интерфейса;

- свертывание иерархий;
- формализация методов;
- замена наследования диктующими и подбор.